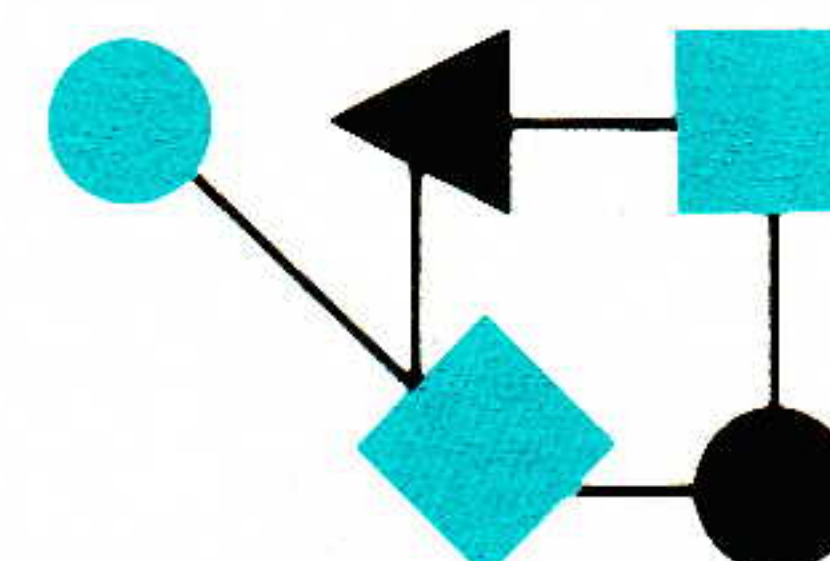


CONNEXIONS



The Interoperability Report

August 1996

Volume 10, No. 8

*ConneXions —
The Interoperability Report
tracks current and emerging
standards and technologies
within the computer and
communications industry.*

In this issue:

HTTP.....	2
Ebone.....	12
Portable computers on the Campus Network.....	20
Statement on Cryptography and the Internet.....	27
Announcements.....	30

ConneXions is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404-1138, USA.

Phone: +1 (415) 578-6900

Fax: +1 (415) 525-0194

E-mail: connexions@interop.com

Subscription hotline: 1-800-575-5717
or +1 610-892-1959

Copyright © 1996 by Interop Company.
Quotation with attribution encouraged.

ConneXions—The Interoperability Report
and the *ConneXions* logo are registered
trademarks of Interop Company.

ISSN 0894-5926

From the Editor

Ten years ago this month Dan Lynch organized the "TCP/IP Implementors Workshop" in Monterey, California. This invitation-only event was a meeting between computer vendors and researchers. Participants included Vint Cerf (co-inventor of TCP/IP), David Clark (The Internet Architect) and Jon Postel (The RFC Editor), to name just a few. Those of us who took part in this the "zero-eth Interop" still remember the buzz surrounding the emerging Internet technologies, the hand-written foils, dinner at the aquarium—and of course the fabulous chocolate-chip cookies.

Five years later, the number of Internet hosts stood at 727,000 and discussions were already underway on the need for a larger address space. A few days ago, the latest Internet Domain Survey from Network Wizards estimated a total of 12 million hosts, an increase of 3 million since January of this year.

The incredible popularity of the Internet is in part due to the introduction of the World-Wide Web in 1993. A recent estimate puts the number of Web pages at some 200,000. An important component of this technology is the *Hypertext Transfer Protocol* (HTTP). This month's first article, by William Stallings, is a tutorial on HTTP.

The growth of the Internet in Europe has been facilitated in part by a number of key network backbone projects. One such network is Ebone, described in *ConneXions* in our May 1993 issue. We asked Frode Greisen, the general manager for Ebone, for an update and an overview of the state of the European part of the Internet.

As portable computers continue to proliferate, the need to connect them to "network outlets" in various locations increases. This is particularly true on university campuses. David Wasley outlines a way in which users on the move can be authenticated and gain access to the campus network. If this system is implemented and standardized it could become very useful for corporate networks as well.

Much debate has surrounded the use of cryptography in computer communications. Security experts all agree that such technology is essential to ensure the integrity of Internet commerce transactions as well as the privacy of individuals. But several governments currently impose restrictions its use through export controls and other measures. The IAB and IESG has issued a statement on cryptography and the Internet. The statement can be found on page 27.

It is not too late to submit proposals for BOF sessions at the upcoming NetWorld+Interop event. You can e-mail your suggestions to ole@interop.com. We are also still seeking volunteers for our *Conference Assessment Team* (CAT). More information about the conference can be found at <http://www.interop.com>.

Back to Basics: **The Hypertext Transfer Protocol (HTTP)**

by William Stallings

Introduction

The *Hypertext Transfer Protocol* (HTTP) is the foundation protocol of the World-Wide Web (WWW) and can be used in any client-server application involving hypertext. The name is somewhat misleading in that HTTP is not a protocol for transferring hypertext; rather it is a protocol for transmitting information with the efficiency necessary for making hypertext jumps. The data transferred by the protocol can be plain text, hypertext, audio, images, or any Internet-accessible information.

This article is based on the most recent (June 7, 1996) specification, HTTP 1.1, Draft 05, which has been forwarded to the IESG for action as a proposed standard.

HTTP overview

HTTP is a transaction-oriented client-server protocol. The most typical use of HTTP is between a Web browser and a Web server. To provide reliability, HTTP makes use of TCP. Nevertheless, HTTP is a “stateless” protocol: Each transaction is treated independently. Accordingly, a typical implementation will create a new TCP connection between client and server for each transaction and then terminate the connection as soon as the transaction completes, although the specification does not dictate this one-to-one relationship between transaction and connection lifetimes.

The stateless nature of HTTP is well-suited to its typical application. A normal session of a user with a Web browser involves retrieving a sequence of Web pages and documents. The sequence is, ideally, performed rapidly, and the locations of the various pages and documents may be a number of widely distributed servers.

Another important feature of HTTP is that it is flexible in the formats that it can handle. When a client issues a request to a server, it may include a prioritized list of formats that it can handle, and the server replies with the appropriate format. For example, a *lynx* browser cannot handle images, so a Web server need not transmit any images on Web pages. This arrangement prevents the transmission of unnecessary information and provides the basis for extending the set of formats with new standardized and proprietary specifications.

Figure 1 illustrates three examples of HTTP operation. The simplest case is one in which a user agent establishes a direct connection with an origin server. The *user agent* is the client that initiates the request, such as a Web browser being run on behalf of an end user. The *origin server* is the server on which a resource of interest resides; an example is a Web server at which a desired Web page resides. For this case, the client opens a TCP connection that is end-to-end between the client and the server. The client then issues an HTTP request. The request consists of a specific command, referred to as a *method*, a URL, and a MIME-like message containing request parameters, information about the client, and perhaps some additional content information.

When the server receives the request, it attempts to perform the requested action and then returns an HTTP response. The response includes status information, a success/error code, and a MIME-like message containing information about the server, information about the response itself, and possible body content. The TCP connection is then closed.

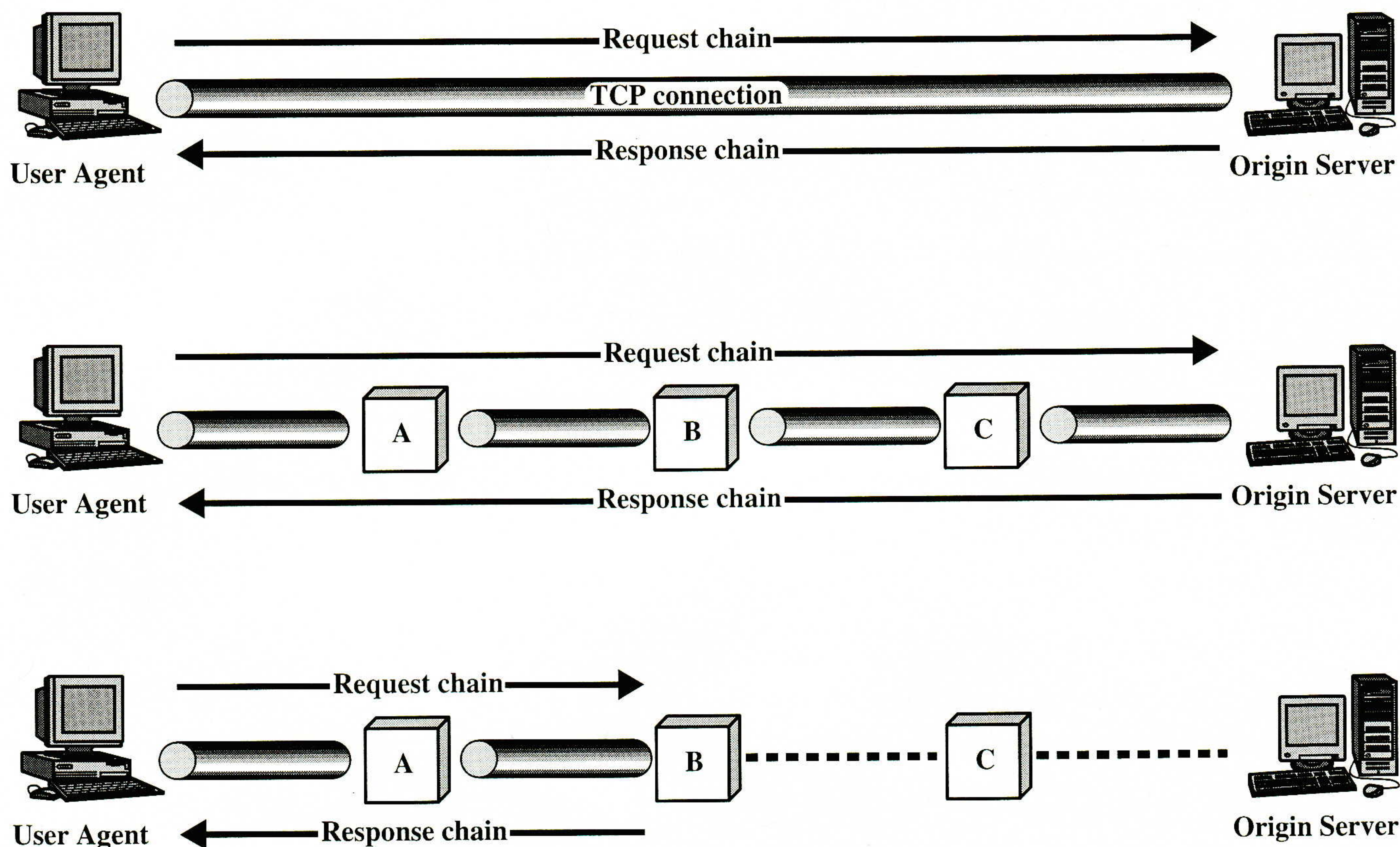


Figure 1: Examples of HTTP Operation

The middle part of Figure 1 shows a case in which there is not an end-to-end TCP connection between the user agent and the origin server. Instead, there are one or more intermediate systems with TCP connections between logically adjacent systems. Each intermediate system acts as a relay, so that a request initiated by the client is relayed through the intermediate systems to the server, and the response from the server is relayed back to the client.

Three forms of intermediate system are defined in the HTTP specification: proxy, gateway, and tunnel, all of which are illustrated in Figure 2.

Proxy

A *proxy* acts on behalf of other clients and presents requests from other clients to a server. The proxy acts as a server in interacting with a client and as a client in interacting with a server. There are several scenarios that call for the use of a proxy:

- *Security intermediary:* The client and server may be separated by a security intermediary such as a firewall, with the proxy on the client side of the firewall. Typically, the client is part of a network secured by a firewall and the server is external to the secured network. In this case, the server must authenticate itself to the firewall to set up a connection with the proxy. The proxy accepts responses after they have passed through the firewall.
- *Different versions of HTTP:* If the client and server are running different versions of HTTP, then the proxy can implement both versions and perform the required mapping.

In summary, a proxy is a forwarding agent, receiving a request for a URL object, modifying the request, and forwarding the request toward the server identified in the URL.

Hypertext Transfer Protocol (continued)

Gateway

A *gateway* is a server that appears to the client as if it were an origin server. It acts on behalf of other servers that may not be able to communicate directly with a client. There are several scenarios in which servers can be used.

- *Security intermediary*: The client and server may be separated by a security intermediary such as a firewall, with the gateway on the server side of the firewall. Typically, the server is connected to a network protected by a firewall, with the client external to the network. In this case the client must authenticate itself to the proxy, which can then pass the request on to the server.
- *Non-HTTP server*: Web browsers have built into them the capability to contact servers for protocols other than HTTP, such as FTP and Gopher servers. This capability can also be provided by a gateway. The client makes an HTTP request to a gateway server. The gateway server then contacts the relevant FTP or Gopher server to obtain the desired result. This result is then converted into a form suitable for HTTP and transmitted back to the client.

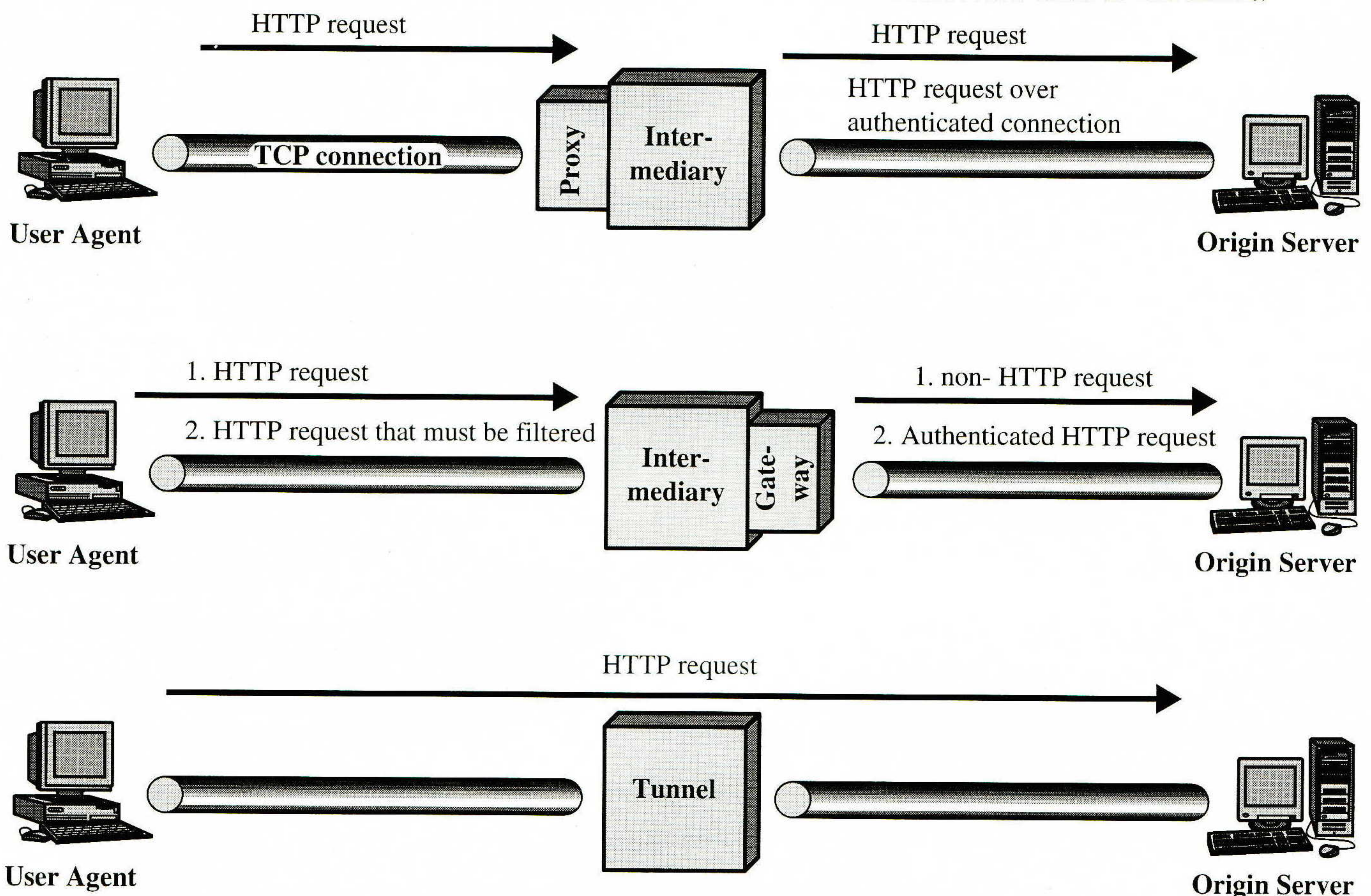


Figure 2: Intermediate HTTP Systems

Tunnel

Unlike the proxy and the gateway, the *tunnel* performs no operations on HTTP requests and responses. Instead, a tunnel is simply a relay point between two TCP connections, and the HTTP messages are passed unchanged as if there were a single HTTP connection between user agent and origin server. Tunnels are used when there must be an intermediary system between client and server but it is not necessary for that system to understand the contents of messages. An example is a firewall in which a client or server external to a protected network can establish an authenticated connection, and then maintain that connection for purposes of HTTP transactions.

Cache

Returning to Figure 1, the lowest portion of the figure shows an example of a *cache*. A cache is a facility that may store previous requests and responses for handling new requests. If a new request arrives that is the same as a stored request, then the cache can supply the stored response rather than accessing the resource indicated in the URL. The cache can operate on a client or server or on an intermediate system other than a tunnel. In the figure, intermediary B has cached a request/response transaction, so that a corresponding new request from the client need not travel the entire chain to the origin server, but is handled by B.

Not all transactions can be cached, and a client or server can dictate that a certain transaction may be cached only for a given time limit.

Messages

The best way to describe the functionality of HTTP is to describe the individual elements of the HTTP message. HTTP consists of two types of messages: requests from clients to servers, and responses from servers to clients. The general structure of such messages is:

```

HTTP-Message = Request | Response
Request = Request-Line
           *(General-Header | Request-Header | Entity-Header)
           CRLF
           [ Message-Body ]

Response = Status-Line
           *(General-Header | Response-Header | Entity-Header)
           CRLF
           [ Message-Body ]

```

With full requests and responses, the following fields are used:

- *Request-Line*: Identifies the message type and the requested resource.
- *Status-Line*: Provides status information about this response.
- *General-Header*: Contains fields that are applicable to both request and response messages, but which do not apply to the entity being transferred.
- *Request-Header*: Contains information about the request and the client.
- *Response-Header*: Contains information about the response.
- *Entity-Header*: Contains information about the resource identified by the request and information about the entity body.
- *Message-Body*: The body of the message.

All of the HTTP headers consist of a sequence of fields, following the same generic format as RFC 822. Each field begins on a new line and consists of the field name followed by a colon and the field value.

Although the basic transaction mechanism is simple, there are a large number of fields and parameters defined in HTTP; these are listed in Table 1.

General header fields

General header fields can be used in both request and response messages. These fields are applicable in both types of messages and contain information that does not directly apply to the entity being transferred.

Hypertext Transfer Protocol (*continued*)

The fields are:

- *Cache-Control*: Specifies directives that must be obeyed by any caching mechanisms along the request/response chain. The purpose is to prevent a cache from adversely interfering with this particular request or response. For example, a server may use this field to inform any caching mechanism en route that it may cache this response for future requests up to a *max-age* time limit.
- *Connection*: Contains a list of keywords and header field names that only apply to this TCP connection between the sender and the nearest non-tunnel recipient. For example, the sender can request a persistent TCP connection (one that remains open beyond the current transaction).
- *Date*: Date and time at which the message originated.
- *Pragma*: Contains implementation-specific directives that may apply to any recipient along the request/response chain.
- *Transfer-Encoding*: Indicates what transformation, if any, has been applied to the message body.
- *Upgrade*: Used in a request to specify what additional protocols the client supports and would like to use; used in a response to indicate which protocol will be used.
- *Via*: Identifies the intermediate protocols and recipients between the user agent and the server.

Request messages

A full request message consists of a status line followed by one or more general, request, and entity headers, followed by an optional entity body.

Request methods

A request message always begins with a Request-Line, which has the following format:

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

The Method parameter indicates the actual request command, called a *method* in HTTP. Request-URI is the URI (*Uniform Resource Identifier*) of the requested resource, and HTTP-Version is the version number of HTTP used by the sender.

The following request methods are defined in HTTP/1.1:

- **OPTIONS**: A request for information about the options available for the request/response chain identified by this URI.
- **GET**: A request to retrieve the information identified in the URI and return it in a entity body. A GET is conditional if the “If-Modified-Since” header field is included, and is partial if a “Range” header field is included.
- **HEAD**: This request is identical to a GET, except that the server’s response must not include an entity body; all of the header fields in the response are the same as if the entity body were present. This enables a client to get information about a resource without transferring the entity body.
- **POST**: A request to accept the attached entity as a new subordinate to the identified URI. The posted entity is subordinate to that URI in the same way that a file is subordinate to a directory containing it, a news article is subordinate to a newsgroup to which it is posted, or a record is subordinate to a database.

- **PUT:** A request to accept the attached entity and store it under the supplied URI. This may be a new resource with a new URI, or a replacement of the contents of an existing resource with an existing URI.
- **DELETE:** Requests that the origin server delete the resource identified by the URI in the Request-Line.
- **TRACE:** Requests that the server return whatever is received as the entity body of the response. This can be used for testing and diagnostic purposes.

ALL MESSAGES			
General Header Fields		Entity Header Fields	
Cache-Control	Transfer-Encoding	Allow	Content-Range
Connection	Upgrade	Content-Base	Content-Type
Date	Via	Content-Encoding	Etag
Pragma		Content-Language	Expires
		Content-Length	Last-Modified
		Content-Location	extension-header
		Content-MD5	
REQUEST MESSAGES			
Request Methods	Request Header Fields		
OPTIONS	Accept	Host	Max-Forwards
GET	Accept-Charset	If-Modified-Since	Proxy-Authorization
HEAD	Accept-Encoding	If-Match	Range
POST	Accept-Language	If-None-Match	Referer
PUT	Authorization	If-Range	User-Agent
DELETE	From	If-Unmodified-Since	
TRACE			
RESPONSE MESSAGES			
Response Status Codes			Response Header Fields
Continue	Not Modified	Length Required	Age
Switching Protocols	Use Proxy	Precondition Failed	Location
OK	Bad Request	Request Entity Too Large	Proxy-Authenticate
Created	Unauthorized	Request-URI Too Large	Public
Accepted	Payment Required	Unsupported Media Type	Retry-After
Non-Authoritative Information	Forbidden	Internal Server Error	Server
No Content	Not Found	Not Implemented	Vary
Reset Content	Method Not Allowed	Bad Gateway	Warning
Partial Content	Not Acceptable	Service Unavailable	WWW-Authenticate
Multiple Choices	Proxy Authentication Required	Gateway Timeout	
Moved Permanently	Request Timeout	HTTP Version Not Supported	
Moved Temporarily	Conflict		
See Other	Gone		

Table 1: HTTP Elements

Request header fields

Request header fields function as request modifiers, providing additional information and parameters related to the request. The following fields are defined in HTTP/1.1:

- *Accept*: A list of media types and ranges that are acceptable as a response to this request.
- *Accept-Charset*: A list of character sets acceptable for the response.

continued on next page

Hypertext Transfer Protocol (*continued*)

- *Accept-Encoding*: List of acceptable content encodings for the entity body. Content encodings are primarily used to allow a document to be compressed or encrypted. Typically, the resource is stored in this encoding and only decoded before actual use.
- *Accept-Language*: Restricts the set of natural languages that are preferred for the response.
- *Authorization*: Contains a field value, referred to as *credentials*, used by the client to authenticate itself to the server.
- *From*: The Internet e-mail address for the human user who controls the requesting user agent.
- *Host*: Specifies the Internet host of the resource being requested.
- *If-Modified-Since*: Used with the GET method. This header includes a date/time parameter; the resource is to be transferred only if it has been modified since the date/time specified. This feature allows for efficient cache update. A caching mechanism can periodically issue GET messages to an origin server, and will receive only a small response message unless an update is needed.
- *If-Match*: Used to make a request conditional. The method in the request should only be performed if the entity identified in this field is present.
- *If-None-Match*: Opposite meaning of "If-Match."
- *If-Range*: Has the meaning, if the entity is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new entity.
- *If-Unmodified-Since*: This header includes a date/time parameter; the resource is to be transferred only if it has not been modified since the date/time specified.
- *Max-Forwards*: Used with the TRACE method, to limit the number of proxies or gateways that can forward the request to the next inbound server.
- *Proxy-Authorization*: Allows the client to identify itself to a proxy that requires authentication.
- *Range*: For future study. The intent is that, in a GET message, a client can request only a portion of the identified resource.
- *Referer*: The URI of the resource from which the Request-URI was obtained. This enables a server to generate lists of back-links.
- *User-Agent*: Contains information about the user agent originating this request. This is used for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations.

Response messages

A response message consists of a status line followed by one or more general, response, and entity headers, followed by an optional entity body.

Status codes

A full response message always begins with a Status-Line, which has the following format:

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

The HTTP-Version value is the version number of HTTP used by the sender. The Status-Code is a 3-digit integer that indicates the response to a received request, and the Reason-Phrase provides a short textual explanation of the status code.

There are a rather large number of status codes defined in HTTP/1.1; these are listed in Table 1. The codes are organized into the following categories:

- *Informational*: The request has been received and processing continues. No entity body accompanies this response.
- *Successful*: The request was successfully received, understood, and accepted. The information returned in the response message depends on the request method, as follows:
 - GET: The contents of the entity-body corresponds to the requested resource.
 - HEAD: No entity body is returned.
 - POST: The entity describes or contains the result of the action.
 - TRACE: The entity contains the request message.
 - Other methods: The entity describes the result of the action.
- *Redirection*: Further action is required to complete the request.
- *Client Error*: The request contains a syntax error or the request cannot be fulfilled.
- *Server Error*: The server failed to fulfill an apparently valid request.

Response header fields

Response header fields provide additional information related to the response that cannot be placed in the Status-Line. The following fields are defined in HTTP/1.1:

- *Age*: Gives the sender's estimate of the amount of time since the response was generated at the origin server.
- *Location*: Defines the exact location of the resource identified by the Request-URI.
- *Proxy-Authenticate*: Included with a response that has a status code of "Proxy Authentication Required." This field contains a "challenge" that indicates the authentication scheme and parameters required.
- *Public*: Lists the non-standard methods supported by this server.
- *Retry-After*: Included with a response that has a status code of "Service Unavailable," and indicates how long the service is expected to be unavailable.
- *Server*: Identifies the software product used by the origin server to handle the request.
- *Vary*: Indicates that the server has selected a representation for this response, rather than query the client.
- *Warning*: Used to carry additional information about the status of the response.
- *WWW-Authenticate*: Included with a response that has a status code of "Unauthorized." This field contains a "challenge" that indicates the authentication scheme and parameters required.

continued on next page

Hypertext Transfer Protocol (*continued*)

Entities	An <i>entity</i> consists of an entity header and an entity body in a request or response message. An entity may represent a data resource, or it may constitute other information supplied with a request or response.
Entity header fields	<p>Entity header fields provide optional information about the entity body or, if no body is present, about the resource identified by the request. The following fields are defined in HTTP/1.1:</p> <ul style="list-style-type: none"> • <i>Allow</i>: Lists methods supported by the resource identified in the Request-URI. This field must be included with a response that has a status code of “Method Not Allowed” and may be included in other responses. • <i>Content-Base</i>: Used to specify the base URI for resolving relative URLs. • <i>Content-Encoding</i>: Indicates what content encodings have been applied to the resource. The only encodings currently defined are compression algorithms. • <i>Content-Language</i>: Identifies the natural language(s) of the intended audience of the enclosed entity. • <i>Content-Length</i>: The size of the entity body in octets. • <i>Content-Location</i>: The resource location for this entity. • <i>Content-MD5</i>: For future study. MD5 refers to the MD5 hash code function. • <i>Content-Range</i>: For future study. The intent is that this will indicate a portion of the identified resource that is included in this response. • <i>Content-Type</i>: Indicates the media type of the entity body. • <i>Etag</i>: An entity tag. • <i>Expires</i>: Date/time after which the entity should be considered stale. • <i>Last-Modified</i>: Date/time that the sender believes the resource was last modified.
Entity body	<p>An entity body consists of an arbitrary sequence of octets. HTTP is designed to be able to transfer any type of content, including text, binary data, audio, images, and video. When an entity body is present in a message, the interpretation of the octets in the body is determined by the header fields Content-Encoding, Content-Type, and Transfer-Encoding. These define a three-layer, ordered encoding model:</p> <pre style="text-align: center;">entity-body := Content-Encoding(Content-Type(data))</pre> <p>The data is the content of a resource identified by a URI. The Content-Type field determines the way in which the data is interpreted. A Content-Encoding may be applied to the data and stored at the URI instead of the data. Finally, on transfer, a Transfer-Encoding may be applied to form the message body of the message:</p> <pre style="text-align: center;">message-body := Transfer-Encoding (entity-body)</pre>
Access authentication	HTTP/1.1 defines a simple challenge–response technique for authentication. This definition does not restrict HTTP clients and servers from using other forms of authentication, but the current standard only covers this simple form.

Two authentication exchanges are defined, one between a client and a server, and one between a client and a proxy. Both types of exchange use a challenge-response mechanism.

Summary

The Hypertext Transfer Protocol (HTTP) is a request-response protocol that can be used on top of TCP to provide a reliable transaction service. HTTP was designed to provide the transfer mechanism between Web browsers and servers.

[This article is based on material in Bill Stallings' *Data and Computer Communications*, Fifth Edition, ISBN 0-02-415425-3, © 1996 by Addison-Wesley. Used with permission. —Ed.]

References

- [1] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1341, June 1992.
- [2] Moore, K., "Representation of Non-ASCII Text in Internet Message Headers," RFC 1342, June 1992.
- [3] Borenstein, N., "A User Agent Configuration Mechanism For Multimedia Mail Format Information," RFC 1343, June 1992.
- [4] Borenstein, N., "Implications of MIME for Internet Mail Gateways," RFC 1344, June 1992.
- [5] Simonsen, K. "Character Mnemonics & Character Sets," RFC 1345, June 1992.
- [6] Crocker, D., "Standard for the format of ARPA Internet Text Messages," RFC 822, August 1982.
- [7] Borenstein, Nathaniel S., "Multimedia Mail From the Bottom Up—or Teaching Dumb Mailers to Sing," *ConneXions*, Volume 5, No. 11, November 1991.
- [8] Vaudreuil, G. , "MIME: Multi-Media, Multi-Lingual Extensions for RFC 822 Based Electronic Mail," *ConneXions*, Volume 6, No. 9, September 1992.
- [9] Crowcroft, Jon and Handley, Mark "The World-Wide Web: How Servers Work," *ConneXions*, Volume 9, No. 2, February 1995.
- [10] Koster, Martijn, "Robots in the Web: threat or treat?" *ConneXions*, Volume 9, No. 4, April, 1995.
- [11] Berners-Lee, T., "A Summary of the WorldWideWeb System," *ConneXions*, Volume 6, No. 7, July 1992.
- [12] Berners-Lee, T., R. Cailliau, A. Loutonen, H. F. Nielsen and A. Secret, "The World-Wide Web," *Communications of the ACM*, Volume 37, No. 8, August 1994.
- [13] Crowcroft, Jon and Handley, Mark, "Problems with the World-Wide Web," *ConneXions*, Volume 9, No. 6, June 1995.
- [14] Mark Handley and Jon Crowcroft, *The World-Wide Web: Beneath the Surf*, ISBN 1-85728-435-6, UCL Press, 1995.

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen professional reference books and textbooks on data communications and computer networking. He has designed and implemented both TCP/IP-based and OSI-based protocol suites on a variety of computers and operating systems, ranging from microcomputers to mainframes. His e-mail address is ws@shore.net and his home in cyberspace is: <http://www.shore.net/~ws/welcome.html>

Ebone and other parts of the European Internet
by Frode Greisen, UNI-C

Introduction

The Internet is growing exponentially in Europe as in the rest of the world. The number of IP hosts registered in the name servers is regularly tracked by the RIPE NCC [1] and by Network Wizards [2]. Table 1 shows the European host numbers along with world-wide numbers and by this measure the European Internet is about 25% of the whole net and the annual growth is around 100%.

Date	Europe	World	Percent of world	Annual Growth
November '90	32	287	11%	—
August '91	73	562	13%	128%
August '92	222	1040	21%	204%
August '93	451	1870	24%	103%
August '94	835	3430	24%	85%
August '95	1774	6800	26%	112%
April '96	2600	—	—	77%

Persons on the core Internet, October 1995:

Persons on the Consumer Internet, October 1995:

6.9 M

26.4 M

Table 1: Number of IP hosts in thousands

IP hosts is one measure. With the onset of commercial applications it also becomes increasingly important to measure the number of persons using the net and various surveys are now carried out by market analysis bureaus. Here we shall only refer to the online survey performed by MIDS [3] which divides users into core Internet users who are those persons that can put information resources such as Web pages and file stores on the net, and consumer Internet users who are those that can only access those resources. Consumer Internet users are those who have full Internet access, just having e-mail does not qualify.

MIDS estimated 26.4 million consumer Internet users in October 1995, and by using scaling and extrapolation factors from Table 1 this gives us around 10 million Europeans on the Internet by June 1996.

It is interesting to compare this to previous expectations. In 1989 the EU supported COSINE project expected the number of researchers using OSI services to be 225,000 by 1992. As we know, OSI never really took off, but in 1992 there were already 222,000 European Internet hosts and referring to the user/machine ratio above probably a somewhat higher number of researchers were using the Internet at that time. So, apart from the fact that a different technology was used the ambitious COSINE prediction was surpassed by reality.

Costs

The basic economic fact determining the shape of the European Internet is the high cost of International lines. Table 2 shows typical costs of 2 Mbps international half links for a number of countries and the US. These numbers are from the *Eurodata Handbook* of July 1995 and actual contract prices today can be somewhat lower, but normally not radically lower. As one can see, the cost of an international E1 line be it to another European country or to the US is around 500,000 Ecu (1 Ecu is around 1.22 USD). It is also apparent that countries who already have competition on the infrastructure such as the UK and Sweden generally have lower costs than the others.

<i>Country</i>	<i>Min cost Europe</i>	<i>Max cost Europe</i>	<i>Cost US</i>
AT	280	373	755
BE	280	377	478
CH	241	385	460
DE	225	361	402
DK	145	299	549
ES	371	371	436
FI 1)	116	303	486
FR	317	415	478
GR	382	382	478
IE	338	360	413
IT	367	367	428
NL	237	285	288
NO 1)	42	148	296
PT	246	375	365
SE	63	268	351
UK	193	239	224
<i>Average</i>	213	295	383

1) excluding local loop

US–Europe half link: 245
 1200 km in US: T1 62, T3 504
 300 km in DK: E1 40, E3 300

Table 2: Cost of 2 Mbps international cable half-circuits in kEcu.
 Border areas not considered. VAT not included.

Also, line costs within a European country—while depending a lot on which country—are generally five to ten times cheaper than international lines for the same distance. In some cases they are comparable to US prices.

Next, there is normally economy of scale both for national and international lines so that e.g., an E1 line may cost 10–13 times a 64 Kbps line and not 32 times more. Also, in the cases where E3 (34 Mbps) international lines are available there is generally some economy of scale, e.g., E3 half lines from Germany cost about ten times more than E1 half links, not seventeen times more.

Finally, rates of the local phone calls necessary for dial-up users to establish their Internet sessions are generally higher in Europe than in the US, and I don't know examples in Europe of the system where unlimited local calls are included in the subscription.

Effects

The effect of these price conditions is firstly that the Internet cannot expand as easily as within the US, secondly that there is a tendency to aggregate traffic within countries and jointly establish the international lines as long as there is economy of scale, i.e., up to E1 and again when approaching E3 load. This means that users buy access from local Internet Service Providers (ISPs) who again buy access from national ISPs who again get access from Pan-European ISPs who then again finally connect to the major backbone ISPs in the US. Some local suppliers by-pass one or two steps in this picture and connect directly to a US backbone supplier, but this normally means a degraded performance for connections to European hosts served by other local providers.

continued on next page

Ebone and the European Internet (continued)

Ebone Ebone was conceived as an interim measure by a number of primarily research networks in 1991. The primus motor in early Ebone development and its first chairman was Kees Neggers of SURFnet and the early developments were described by Bernhard Stockman [4]. For Ebone news check [5].

When the EU supported Dante [13] services were established around 1993, a number of research networks left Ebone but other members, primarily RENATER in France, Aconet in Austria and ECRC in Germany wanted to continue, and as can be seen from Table 3 Ebone is in a sound growth.

	Mid-1992	1993	1994	1995	1996
Members	18	30	28	50	71
PoPs	5	6	3	5	6
Access/Mbps	4	6	8	16	41

Table 3: Ebone history

Ebone is a consortium of around 70 members in 27 countries, i.e., there is as yet no legal body called Ebone. This construction provides the framework for cooperation between competing and very diverse members including research networks, Public Network Operators (PNOs) and other commercial ISPs.

Mission The Ebone mission is simply to provide global Internet access for its members and this is done by building a backbone between the Ebone Points of Presence (*Ebone Boundary Systems* or EBSs as we call them), by connecting that backbone to major US backbone providers—at this time SprintLink—and by establishing peering relations with other pan-European ISPs. Note that another branch of Sprint, Sprint International is also an Ebone member but this is entirely distinct from the Ebone connection to SprintLink.

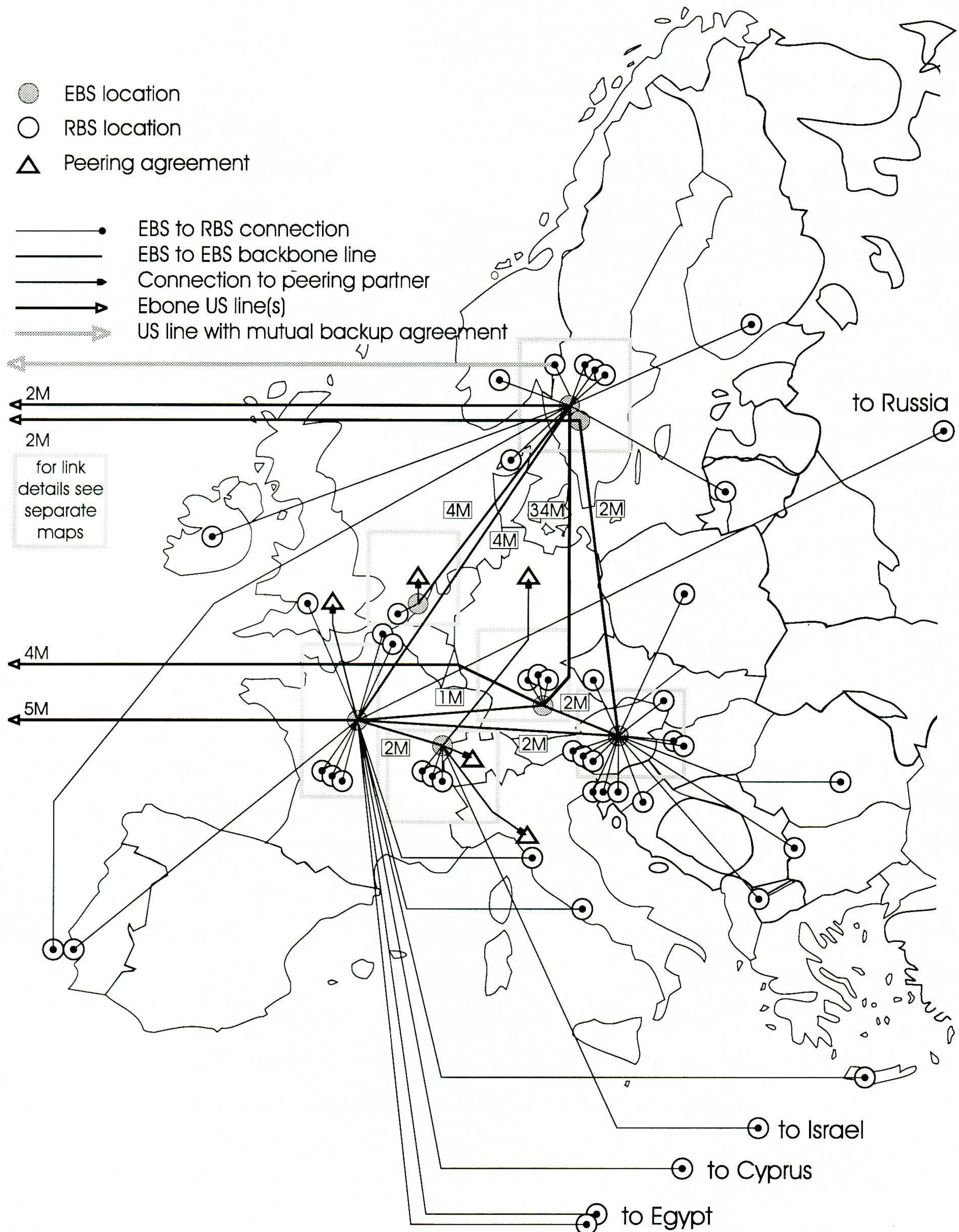
Finances Since there is no Ebone company, contracts for the common resources, primarily the backbone lines and routers, are with some of the members. The financial clearing is done via the same contract type for all members: The member buys a certain access bandwidth at an EBS at common rates calculated to make the overall budget balance and if the member provides a common resource then the cost of that resource is subtracted from the member’s payment which may thus turn negative, i.e., become a refund. This way some members pay to the clearing house which then transfers funds to the members that provide the common resources.

The cost of the member’s line to access an EBS is not considered a common resource and is carried by the member. Due to this construction, it is difficult to show the development of Ebone’s turnover, but we can say that Ebone is a not-for-profit consortium, and as the cost of an European international E1 is around 500 kEcu the total cost of Ebone’s backbone is around 10 MEcu according to the diagram on the right at this time.

The present backbone is shown on the next page. Several upgrades are planned and expected during the summer, including the re-establishment of an EBS in London.

Ebone Configuration

per May 31, 1996



Ebony and the European Internet (*continued*)

Members

The present Ebony membership is listed in Table 4. Talks are ongoing with other organizations and while Ebony was started in Europe and while we do not expect to be a major US network there is no specific wish to limit Ebony to a particular geographic region and the establishment of a US PoP is being investigated. Ebony's initial members were primarily research networks but there has never been an appropriate use policy limiting its use and today the majority of the members are commercial organizations.

ACONET (AT)	FORTH (GR)	Prolink (CH)
ANET (AT)	FRCU (EG)	Renater (FR)
APA (AT)	Frontier Comm. (UK)	RNC (RO)
ARGE (AT)	GAMS (AT)	SANET (SK)
ARNES (SI)	HEANET (IE)	Schibsted Net (NO)
AT-Net (AT)	HP (CH)	Sprint Internatinal (US)
BT (UK)	HUNGARNET (HU)	SwipNet (SE)
Cable Internet (UK)	ILAN (IL)	Taide (LT)
CARNet (HR)	INS (AT)	Telecom Austria (AT)
CERN (CH)	Internet-Way (FR)	Telecom Eireann (IE)
CESNET (CZ)	Internet Egypt (EG)	Telecom Italia (IT)
CINECA (IT)	Internet Egypt (EG)	TeleDanmark/UNI-C (DK)
Croatian PTT (HR)	Interpac (BE)	Telecom Slovenia (SI)
CSO (AT)	IP Global (PT)	Telia (SE)
CWIX (US)	K2.net (SI)	Tuvaka (TR)
Cyberlink (CH)	MARNet (MK)	Transpac (FR)
CYNET (CY)	MSU (RU)	Transpac (SE)
DATANET (FI)	NASK (PL)	Unicom (BG)
Datanet (HU)	Netway (AT)	Unisource (SE)
Demon (UK)	Nice Tech. (FR)	VAM (AT)
ECRC (DE)	NORDUnet	ianet (AT)
Eunetcom (DE)	Pan Amp (DE)	XLINK (DE)
EuroTel (SK)	Plus Comm. (AT)	Xpoint (AT)
FCCN (PT)	Pressimage (FR)	

Table 4: Ebony member organizations

Due to the high cost of European lines members normally only buy the access capacity they need. In addition nearly all Ebony members are ISPs who aggregate traffic from large numbers of organizations and individuals so access lines are fully loaded. Therefore a principal Ebony policy is no overbooking so the capacity of the backbone lines from each EBS should be as large as the sum of accesses to the EBS. Combined with the high cost of international lines this makes Ebony's access rates much higher than the rates of US ISPs, but on the other hand, Ebony will secure its members good access to both the American and the European part of the Internet. So far, connection to Asia is by transit via the US.

The European Internet scene

With the growth and commercialization of the Internet it becomes increasingly difficult to establish how the infrastructure that carries the packets is built. ISPs show what they want to show in their press releases and on their Web pages, and it may sometimes be hard to distinguish facts from plans and intentions.

With these reservations, an attempt to describe the European situation is as follows: If we describe pan-European ISPs as organizations or groups that provide a backbone for ISPs in most of the European countries, then we have Ebony, Dante, EUnet and perhaps also Pipex, now UUNet Europe and thus part of Alternet/MFS. A comparison of key factors is shown in Table 5.

	EUnet	Ebone	Dante	Pipex
<i>Countries</i>	41	27	22	15
<i>Subscription / Mbps</i>	—	41	36	—
<i>US access / Mbps</i>	12	15	20	7

Table 5: Major pan-European IP backbones

Note that development is fast, so the measures above may well have changed by the time you read this article.

Basically, these networks all have backbones built by 2 Mbps (E1) lines as the next step which is 34 Mbps (E3) in Europe is still very expensive if available at all. Dante has coordinated an EU supported project to connect the national research networks with 34 Mbps line since Spring 1995 and one may expect that perhaps lines are put into operation by the end of 1996. In the mean time, the Swedish supplier Tele2 in cooperation with Sprint introduced a 34 Mbps Transatlantic line from Stockholm primarily for NORDUnet in July last year, next NORDUnet put in operation a 34 Mbps line between Helsinki and Stockholm in March and most recently Ebone opened 34 Mbps line between Stockholm and Munich in May.

Major infrastructure players

In addition to these four groups, other large organizations have announced plans or are expected to establish a pan-European infrastructure: Global One, the subsidiary of Deutsche Telecom, France Telecom and Sprint, Concert, the consortium of British Telecom and MCI, and Uniworld, the partnership of AT&T and Unisource which again is a subsidiary the government owned PNOs in Sweden, The Netherlands, Switzerland and Spain. In all three cases we have a major US supplier in alliance with large European PNOs, but it's not always clear if the focus of the consortium is on telephony, private data networks, cable provision or on acting as an Internet Service Provider.

However, there are several European PNOs and emerging infrastructure providers in the gradually liberalized European telecommunications market that are not part of the three big alliances. The PNOs will want to establish Internet services in their home countries and the infrastructure providers (railways, power companies and cable operators) are ready to provide the physical layer.

It should be remembered that the establishment of a European Internet infrastructure is still a multi-language, multi-regulation, multi-currency, multi-million venture even if limited to 2 Mbps lines.

It is estimated that there are between 2,000 and 4,000 ISPs worldwide and for Europe one can assume that the number is close 417 which is the number of Internet registries listed by RIPE. Now, Dante, EUnet and Pipex have around one member per country, Ebone has 66 members totaling around 150 ISPs served by the four backbone providers listed in Table 5. So, where do the rest connect ?

Well, some are connected to the customers of EUnet, Dante, Pipex and Ebone. Others connect directly to US providers.

What all IPSs want to offer to their customers is global Internet access, so they require this from their backbone provider. The backbone providers in turn need to make sure they can provide this and accordingly they need to cooperate.

Ebone and the European Internet (*continued*)

Unlike the US, Europe has had no NSF to establish NAPs nor a CIX so the European backbone providers have had a number of options:

- To establish direct connections with each other.
- To peer via one or more of the emerging International Internet exchanges, e.g., in Stockholm, Amsterdam, London, Paris, Geneva—or in Washington, DC.
- To hand over the problem of last resort routing to a major US provider such as Altnet, Sprint or MCI.

As in the rest of the world, European providers have the difficulty of deciding when they see other ISPs as suppliers to pay to, as peers to have no-settlement peering with, or as customers who should pay. There is probably no consistent and objective formula worked out yet to solve this problem anywhere in the world so decisions are taken from what each provider finds is his best interest. At present all three solutions above are used for traffic interchange between European ISPs. It should be noted that the third solution may be more expensive than the others and that it basically defines the US providers as the top level providers.

The future

The major factor that will effect the European Internet is positive: By January 1, 1998 most EU countries will allow full competition on telecommunication services and infrastructure. Some countries, e.g., Germany and France and Denmark are even expected to improve the date to July 1, 1996. Experience from the US, the UK, Sweden and Finland who have competition today show that this will dramatically decrease line costs and several companies such as railways and power suppliers already have alternative infrastructure to offer as soon as allowed.

However, the supply of Transatlantic cables capacity does not seem to be abundant and provision of new cables takes several years. Thus, prices on intercontinental connections can be expected to stay high which indicates that the US and Europe Internet markets will not fully merge very soon.

Another set of legislations or regulations may affect the European Internet. Only a few years ago the European Union and the European Industry viewed the Internet as only an experiment, not as a serious contender for the Information Society infrastructure: In 1994, the Bangemann report [6] only noted that the Internet was large but had serious security problems and that Europe should follow it closely, while the Round Table of large European Industry [7] stated that the Internet was using old technology and was run by volunteers. Now, governments in Europe and worldwide see the Internet as a reality and are considering regulation in matters of connectivity, contents, copyright and cryptography. In particular, if the providers do not sort out their peering relations in a way that is perceived as fair and open to competition, then the Internet is now so large that Governments may act to solve the problem.

Finally, what will really make governments think is the prospect of losing Value Added Tax (VAT) income. If suppliers like CompuServe can sell information access technically delivered in the US to Europeans without adding VAT then it does not take much of imagination to see that large amounts of lost taxes may occur when electronic trade becomes a significant part of the population's spending.

The notion of a bit tax to be levied from IPSs has already been ventilated by tax consultants of the EU. Internet activists should be alert and act in time.

Conclusion

All taken into account, it's clear that the Internet in Europe will continue to grow. International lines will now fall in price but and especially Transatlantic lines will continue to be expensive so there will be a continued aggregation of international Internet traffic. As there is a general strong political will to break down telecommunications monopolies, it's unlikely that one or two suppliers will be allowed to dominate the European Internet so there will be a continued strong competition between the European suppliers mentioned above plus probably IBM, AOL and others. Ebone plans to be part of this.

References

- [1] `gopher://info.ripe.net/ripe/Houstcount/History/`
- [2] `http://www.nw.com/zone/WWW/top.html`
- [3] *Matrix News*, Volume 6, No. 2, February 1996.
- [4] *ConneXions*, Volume 7, No. 5, May 1993, "Special Issue: Focus on Europe."
- [5] `http://www.ebone.net`
- [6] `http://ww2.echo.lu/eudocs/en/report.html`
- [7] European Round Table of Industrialists: "Building the Information Highways," June 1994.
- [8] Crowcroft, J., "A Brief Glossary/Overview of the European Internet," *ConneXions*, Volume 9, No. 12, December 1995.
- [9] Stockman, B., "Current Status on Networking in Europe," *ConneXions*, Volume 5, No. 7, July 1991.
- [10] Stockman, B., "EBONE, The European Internet Backbone," *ConneXions*, Volume 7, No. 5, May 1993.
- [11] Stockman, B., "Global Connectivity: The Global Internet Exchange (GIX)," *ConneXions*, Volume 7, No. 11, November 1993.
- [12] Réseaux Associés pour la Recherche Européenne, *ConneXions*, Volume 6, No. 1, January 1992.
- [13] Bersee, J., "Profile: DANTE and EuropaNET," *ConneXions*, Volume 8, No. 6, June 1994.
- [14] Kowack, G., "Profile: EUnet," *ConneXions*, Volume 7, No. 11, November 1993.
- [15] Karrenberg, Daniel, "The RIPE NCC and the Routing Registry for Europe," *ConneXions*, Volume 7, No. 11, November 1993.

FRODE GREISEN is working with the Danish Computing Center for Research and Education (UNI-C) as a chief consultant in networking. He was the president of the European Research and Education Network (EARN) from 1989 to 1995. At the merger between EARN and Réseaux Associés pour la Recherche Européenne (RARE) in October 1994 into the Trans-European Education and Research Networking Association (TERENA) he was elected president of TERENA and he served in that position until May 1995. He is a member of the Board of Trustees of the Internet Society and is serving as its treasurer since 1992. Since 1992 he is acting as general manager of Ebone, a European IP backbone connecting 55 Internet Service Providers in 27 countries. Frode Greisen obtained his M.Sc. in electrical engineering in 1964 and a Ph.D. in solid state physics in 1968 from the Technical University of Denmark. He has worked with computers since 1963 and with computer networks since 1984. E-mail: `Frode.Greisen@uni-c.dk`

Authenticating Aperiodic Connections to the Campus Network

by David L. Wasley, University of California

Introduction

The demand for dial-in access to our campus networks is likely to be exceeded by demand for direct connections from personal portable computers. Well over 50% of UC Berkeley students own their own computers and an increasing number of these are portables. In anticipation of this trend, the newest campus lecture halls and library study areas are wired for network connections at every seat. However, network service is not yet offered at these locations. Unlike dial-in access, there is no widely available, convenient mechanism for authenticating the individual who might attach their computer aperiodically to a part of the campus network. This article suggests a way to use existing technologies, with minor extensions, to solve this looming problem. If successful, it might change significantly the way large parts of our campus networks are managed.

Context

There was no need for authentication at the central modem hub when campus modem services were first established. Data flow was character oriented and the only destinations available were local time-sharing systems which performed their own authentication. With the advent of serial line Internet protocols (SLIP and PPP [1]), it became possible to establish the dial-in user's computer as a node on the campus network. This new type of service allowed access to all nodes on the campus network and, unless specific measures were taken, allowed access to the entire Internet. In response to concern over anonymous use of campus resources as well as the potential for abuse of systems on the Internet, campuses implemented authentication as the first step in establishing a network connection via a dial-in modem. Commercial Internet Access Providers rely on this step to account for use of their fee-based services.

LAN technologies, on the other hand, developed in a highly controlled environment where connections were established permanently to computers owned by known individuals or departments. Authentication was not an issue. The network was simply a resource available to all campus computers. It is still the case that LAN technologies do not offer a mechanism that would support an authentication step as link level connectivity is being established.

The idea of a "live network outlet on the wall" to which anyone's computer can be attached is not new. It has long been an appealing idea in areas where the population is mobile, such as a classroom or reconfigurable office. One of the requirements for widespread use of this type of service—automatic configuration of the computer's network protocol parameters—is embodied in the *Dynamic Host Configuration Protocol* (DHCP) [2]. What must be added to a DHCP implementation is an authentication step supported by enhanced intelligence in network hubs serving the "live network outlets."

Goals of the Authentication Model

In order to be truly useful, the authentication mechanism must fit easily into practical environments and should complement other network support requirements. A mechanism has been designed to achieve the following goals:

- To identify a straight forward mechanism to support authentication at the time a link level connection is established;
- To incorporate this as part of an automatic configuration process to ensure a high probability that the connection will function properly;

- To deny service unless authentication is completed and then to provide service only to the authenticated user's computer;
- To minimize the potential for damage to the network by unauthorized connections;
- To use existing technologies, including authentication services, as much as possible, and to be adaptable to new ones;
- To minimize any manual intervention required to support the service.

These goals are meant to address the most general case of ensuring that basic LAN connections are enabled only for legitimate users of our network. It appears that the goals above can be achieved with fairly simple extensions to standard technologies available now for the components shown in Figure 1.

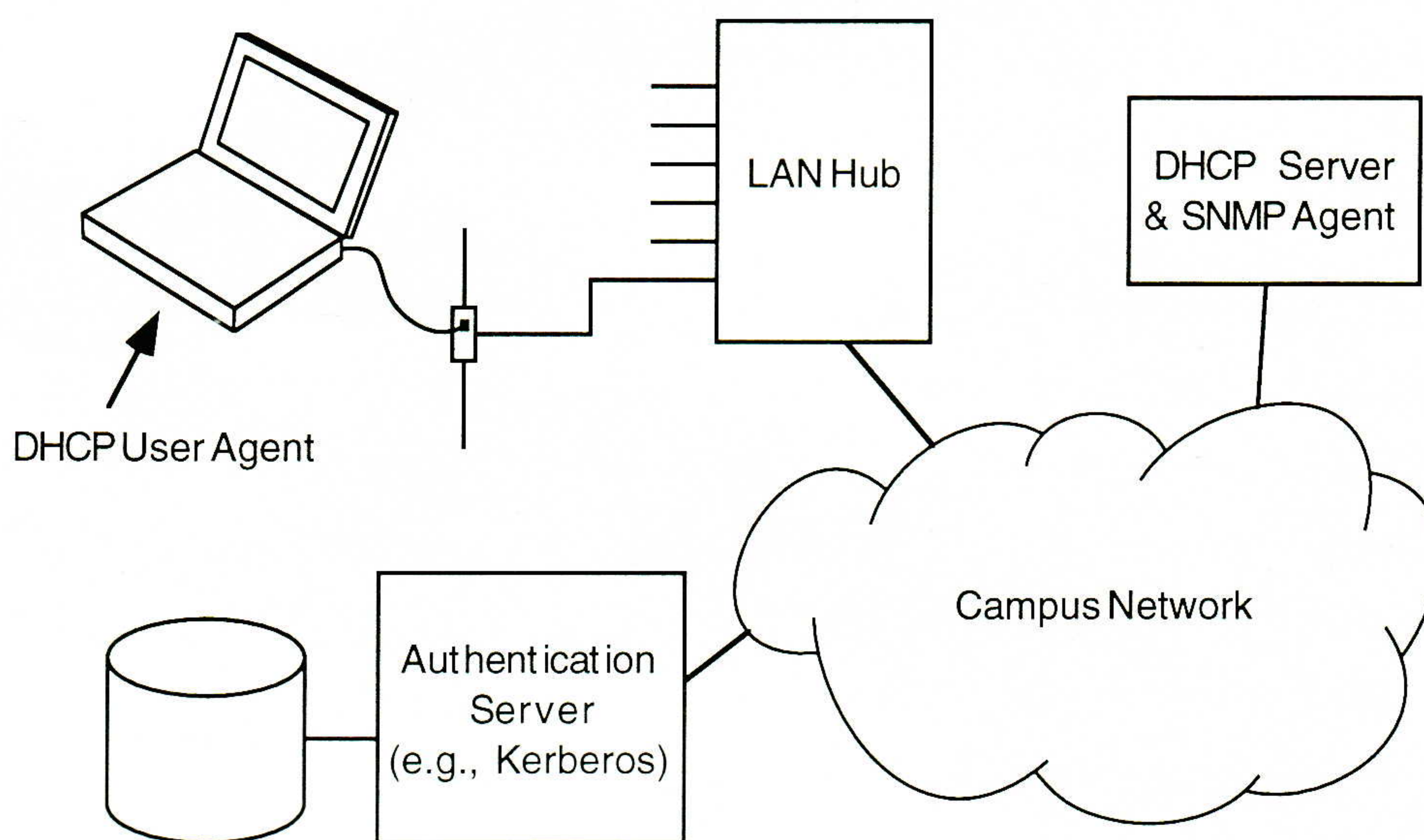


Figure 1: Components of the Authenticated DHCP Model

Constraints on the Model

There are two basic limitations inherent in the requirement to control access. First, there must be a one-to-one mapping between a LAN hub port and an associated user which therefore constrains the use of secondary work group (non-intelligent) hubs. Second, protocols such as DECnet that change MAC addresses will not be usable without a rather significant extension of the protocol since the authentication must be bound to a particular MAC address as the only way to recognize a particular computer at the link level.

Design considerations

Many considerations went into the design of the proposed authentication mechanism. It seems clear that the LAN hub must provide several important functions that can not be done easily elsewhere. For example, the hub must be able to detect the presence of a computer that is attempting to gain access and the hub must recognize the link level identity of that computer.

The simplest way to recognize a newly connected computer would be to receive *from* the computer a packet intended to initiate a configuration and authentication process. However, the hub while in the idle state must not transmit any usable packets *to* the computer since that would allow "eavesdropping." Therefore the LAN hub port must be able to be put into an asymmetric state except when an authenticated connection is present.

continued on next page

Authenticating Aperiodic Connections (*continued*)

Unfortunately the ability of the attached computer to send packets opens the possibility of abuse, e.g., flooding or jamming the network. Therefore some means must be available to shut down the LAN hub port entirely when necessary.

With current products, there is no reliable and practical way for a remote process to associate a MAC address with the particular LAN hub port. Even if the LAN hub's MIB includes the MAC address associated with each port, it would be necessary to scan the entire network segment in order to find it. Polling LAN hubs from a remote authentication server would not detect the MAC address reliably. It would be much more straight forward for the LAN hub controller under specific conditions to offer that information directly.

Finally, a given port must be switched reliably to an unusable state when no longer in use by an authenticated user. Merely the capability for remote on/off control by an SNMP [3] agent is not enough. Consider a connection that has been established properly and is in the "usable" state: there is no reliable way to know if the attached computer is removed and another one inserted in its place unless the LAN hub recognizes either the loss of "carrier" or a different MAC address received from the attached computer. Loss of "carrier" alone is not enough if the connection was made via a secondary LAN hub.

Fortunately LAN hub products exist today that can support most of the required functionality described above. Both so-called "data secure" LAN hubs and switching LAN hubs have the ability to recognize and act upon the MAC address of an attached computer. It is reasonable to assume that future hub products could provide equivalent functionality, including the minor extensions required for reliable access control.

An important design decision was whether authentication would take place over basic link level communication or would use full network level transport. It is undesirable in a large distributed environment to build an authentication process into the hub itself since this creates yet another system requiring operational support. It might be possible to build enough intelligence into the hub to perform a link level handshake with the attached computer and then verify the user with a separate authentication service. However, that would require a new set of protocols and software and a much more complex hub. Therefore I have chosen to assume full network level transport will be used.

An implication of the above is that the newly connected computer must be able to communicate at least to the local subnet in order to reach an authentication service. In larger installations it will be most desirable to be able to reach a campus-wide authentication service. This implies that there be a relay server on each subnet, similar to the BOOTP [4] relay but in support of the authentication service, or else the protocol used by the computer must be routable—i.e. it must be IP. Rather than require another type of server on every subnet, the proposed design requires that the user's computer be properly configured and that the hub allow two-way communications before authentication can be initiated.

Finally, since there is assumed to be no pre-registration of the computer with the authentication service, we can not assume that secure communication at the link level is possible. In most environments we do not want to send the new user's password "in the clear" over a network with arbitrary users on it.

Therefore we must consider use of an authentication system such as Kerberos [5] that addresses that concern. Again, this will require that the new user's computer be properly configured and be able to exchange IP packets with the secure authentication server.

A Conceptual Model

The proposed model for authenticating network connections assumes the use of standard DHCP for configuration of the computer (end-station) and some form of standard authentication service such as Kerberos. The model has three main components: an enhanced DHCP user agent, an intelligent hub that supports disabling and enabling of individual ports, and a DHCP server that requires the authentication step to be completed. Most of the detail of each component exists today. This article will suggest what needs to be added in order to make a complete system that will serve a broad range of needs on our campuses.

An important element of any authentication requirement is the particular authentication mechanism. Ideally anyone who is authorized to use services on our campus network should be able to connect to a LAN outlet *a priori*. Many campuses are running or are in the process of implementing some version of the Kerberos technology under which every member of the campus community will have an identity. Therefore Kerberos was chosen to illustrate the use of an authentication server in the proposed model. Other technologies could be used if appropriate.

Enhanced DHCP User Agent

The DHCP software in the user's computer would combine standard DHCP with standard Kerberos initialization and service authorization. The only "enhancement" would be to combine both functions in a way that would appear seamless to the user.

A related issue beyond the scope of this article is the integration of Kerberos support into the Mac and PC desktop environment. For some desktop environments it might not be possible yet to use Kerberos for more than the connection authentication.

Intelligent Hub with Port Control

Certain LAN hubs today include hardware to recognize the MAC address of an attached machine and suppress data not destined for that machine. One class of these are sometimes referred to as "secure hubs" since they prevent "eavesdropping" on the network by obscuring the data field in all packets except those destined for the attached computer. A rapidly growing class of such hubs are the so-called "LAN switching hubs" that act as multi-way filtered repeaters, often with multiple internal paths. Switching hubs only direct packets to a port if they are destined for the attached computer.

This type of capability also could be used to completely deny service to an attached computer. Rather than automatically "learning" the MAC address of the attached computer, the hub would allow a MAC address to be "set" for each port by means of an SNMP command. To disable a port, a MAC address that is randomly selected or impossible in real hardware would be "set" by the DHCP server (or its SNMP agent). To enable a port, the actual MAC address of the attached computer would be "set." An important benefit of this approach to LAN hub port control, as opposed to simple "on/off" control, is that it prevents "eavesdropping" even after a connection is authenticated.

In addition to this capability, the LAN hub should generate an SNMP **Alert** message if the MAC address of a received packet differs from that "set" for any port. This particular alert would be forwarded to a specific SNMP server address preconfigured into the hub.

Authenticating Aperiodic Connections (*continued*)

Finally, the LAN hub should send an **SNMP Alert**, and optionally disable the port, when link level connectivity is lost, e.g., the attached computer is unplugged or powered off.

None of these capabilities require new hardware in today's advanced LAN hubs. All are simple programmed extensions of existing capabilities. However, such extensions must be made by the hub vendor.

DHCP Server with Authentication Support

The most complex component is the enhanced DHCP server. In addition to the standard DHCP protocol, it must understand the authentication method to be used and interact properly with the user agent and the authentication server. If Kerberos is used, it must be prepared to accept a ticket for DHCP Services and correlate it with a previously received standard DHCP request. It also must maintain a timer for each port that is in the initialized, pre-authorization state.

The DHCP server also must interact with an SNMP agent to control the state of the LAN hub ports. This interaction could be simplified by including the SNMP code in the DHCP server itself. In any case, a certain amount of additional state must be maintained for each port but the critical state must be known for only a short period.

The DHCP server or SNMP agent could also provide protection against accidental or intentional "denial of service" attacks such as flooding or jamming of the local subnet. Since an **SNMP Alert** is generated for each packet received from a non-authenticated computer, a flood of such packets over a short period could be detected and the associated LAN hub port disabled completely for a suitable period.

Required development would include adding the timer function, the SNMP agent, and the authentication interface to existing DHCP server code.

How it would work

In operation, the following scenario would take place:

- 1 The initial state of the network outlet, as determined by the LAN hub port settings, is to be able to receive packets from an attached computer (end station) but to transmit nothing, or at most unusable packets, to the end station. This might be achieved by use of a "secure hub" or switching hub that suppresses outgoing data not destined for a predefined MAC address.
- 2 Once plugged in and powered up, the first operation a user performs is to execute an enhanced *DHCP User Agent* (DUA) on their computer. This DUA begins by broadcasting a standard DHCP configuration request.
- 3 Two things happen then, more or less in parallel:
 - 3A The LAN hub sees activity on the idle port and sends an **SNMP Alert** to the designated *SNMP Monitor Agent* (SMA)—which should be on the same platform as the DHCP server. The LAN hub does not enable the port. The alert message contains the port number and the MAC address. (This is probably new functionality for commercial LAN hub products.)
 - 3B A DHCP server receives the configuration request, perhaps forwarded by a DHCP proxy on the local subnet. This configuration request must contain the MAC address of the requesting end station.

4 The SMA forwards the MAC address and port ID of the end station to the DHCP server in a request for permission to enable the port. This step is necessary in case the authentication fails (see step 7 below).

5 The DHCP server:

5A Caches the parameters and acknowledges to the SMA the request to enable the port. The SMA then directs the LAN hub (using authenticated SNMP “sets”) to allow bidirectional communication for that MAC address on that port.

5B Returns configuration parameters to the end station in the standard way, including its own IP address in case there are multiple DHCP servers that might have responded.

5C Sets a timer, associated with the assigned IP address, that defines how long it will wait for confirmation of an authentication step. This parameter must be configurable and will be longer for large networks than for small ones.

6 Once the DUA on the end-station configures the IP protocol stack, it initiates the authentication step:

If Kerberos is used, the DUA initiates the Kerberos authentication and saves the *Ticket Granting Service* (TGS) ticket for further use by the user. It then uses the TGS ticket to request a ticket for “DHCP Service” from the actual server that responded to the configuration request. When it receives this ticket, it sends it to the DHCP server.

Other authentication mechanisms might be used, such as TACACS, with appropriate hooks in the DHCP server and DUA.

7 Once the DHCP server is satisfied with the authentication step, e.g., by receiving a valid Kerberos “DHCP Service” ticket, it clears the timer for that end station IP address. If the timer expires, the DHCP server notifies the SMA to reset the LAN hub port associated with that configuration request.

8 The LAN hub port is reset to the initial (unusable) state if:

8A the timer for an end station expires before authentication is completed, or

8B the LAN hub detects “loss of carrier” e.g., loss of link level integrity on a 10BaseT connection, or

8C the SMA receives an alert from the LAN hub that a packet with a different MAC address has been received on an operational port. The LAN hub may have no way to distinguish between an operational port (configured with a valid MAC address) and an idle port (with an invalid MAC address). Therefore, the state for each port may have to be kept in the SMA. The SMA could choose to require re-authentication if it receives such an alert as an alternative to shutting down the port.

The scenario above is probably adequate for use in a campus environment but it is not foolproof. Various attacks might be made to disable the SMA or inject illicit traffic during the authentication window. Most LAN hubs already will protect against jamming of the network by automatically shutting down that port. The SMA could be programmed to notice less severe flooding of the network from an unauthorized station and could completely shut down that port.

Authenticating Aperiodic Connections (*continued*)

In any case, I believe that any remaining vulnerabilities are no worse than we face today with many of the existing connections on our networks.

The use of Kerberos in the scenario above has an important secondary advantage: we want a universal, strong authentication mechanism in widespread use on our networks and we want it to be "user friendly." By incorporating Kerberos initialization into the required setup stage, we not only achieve the goals of an authenticated link level connection but we also prepare the user for access to Kerberized campus resources.

If this scenario is implemented well, with reliable and well managed servers, then it could be used to initialize almost all workstation connections on the campus network. The configuration and authentication steps would be part of booting any machine, permanent or mobile. This would ease the problems of relocating or changing machines, adding nodes to the network, etc. Of course, this would require a well thought out scheme for address allocation and reclamation. For reliability, redundant servers would be needed with appropriate synchronization. Scenarios such as recovering from a wide spread power failure should be considered. However, a significant advantage of this strategy is that, by logging authenticated DHCP requests, the campus Network Operations group would have a far more accurate picture of the real connections on the network than they do today with manual procedures.

Conclusion

Certainly there are other models for achieving the goals defined for this service. What this article has tried to illustrate is that it can be done in a straight forward way with technology that, for the most part, already exists. Several LAN hub manufacturers already have expressed interest in developing a prototype of the firmware modifications. I hope to find similar interest in adapting DHCP implementation software to include an authentication step.

If the prototype proves successful, I would insist that the specifications be clearly defined and freely available. I believe we would all benefit from a readily available common solution to this widely shared problem.

References

- [1] W. Simpson, Editor, "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994.
- [2] R. Droms, "Dynamic Host Configuration Protocol," RFC 1541, October 1993.
- [3] M. Schoffstall, M. Fedor, J. Davin, J. Case, "A Simple Network Management Protocol (SNMP)," RFC 1157, May 1990.
- [4] W. Croft, J. Gilmore, "Bootstrap Protocol," RFC 951, September 1985.
- [5] J. Kohl, B. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, September 1993.

DAVID WASLEY holds a Masters Degree from the University of California, Berkeley, and has been a member of the staff of the University for 27 years. For the last decade he has been responsible for the development of the UC Berkeley campus data network and associated services. He was active in the founding and development of the Bay Area Regional Research Network (BARRNet). He is co-author of RFC 1709 "K-12 Internetworking Guidelines." Recently he joined the UC Office of the President in order to focus on new issues and challenges in the area of Information Infrastructure Planning. E-mail: David.Wasley@UCOP.EDU

IAB and IESG Statement on Cryptographic Technology and the Internet

July 24, 1996

Introduction

The *Internet Architecture Board* (IAB) and the *Internet Engineering Steering Group* (IESG), the bodies which oversee architecture and standards for the Internet, are concerned by the need for increased protection of international commercial transactions on the Internet, and by the need to offer all Internet users an adequate degree of privacy.

Security mechanisms being developed in the Internet Engineering Task Force (IETF) to meet these needs require and depend on the international use of adequate cryptographic technology. Ready access to such technology is therefore a key factor in the future growth of the Internet as a motor for international commerce and communication.

The IAB and IESG are therefore disturbed to note that various governments have actual or proposed policies on access to cryptographic technology that either:

- (a) impose restrictions by implementing export controls; and/or
- (b) restrict commercial and private users to weak and inadequate mechanisms such as short cryptographic keys; and/or
- (c) mandate that private decryption keys should be in the hands of the government or of some other third party; and/or
- (d) prohibit the use of cryptology entirely, or permit it only to specially authorized organizations.

We believe that such policies are against the interests of consumers and the business community, are largely irrelevant to issues of military security, and provide only a marginal or illusory benefit to law enforcement agencies, as discussed below.

The IAB and IESG would like to encourage policies that allow ready access to uniform strong cryptographic technology for all Internet users in all countries.

The IAB and IESG claim:

The Internet is becoming the predominant vehicle for electronic commerce and information exchange. It is essential that the support structure for these activities can be trusted.

Encryption is not a secret technology monopolized by any one country, such that export controls can hope to contain its deployment. Any hobbyist can program a PC to do powerful encryption. Many algorithms are well documented, some with source code available in textbooks.

Export controls on encryption place companies in that country at a competitive disadvantage. Their competitors from countries without export restrictions can sell systems whose only design constraint is being secure, and easy to use.

Usage controls on encryption will also place companies in that country at a competitive disadvantage because these companies cannot securely and easily engage in electronic commerce.

Escrow mechanisms inevitably weaken the security of the overall cryptographic system, by creating new points of vulnerability that can and will be attacked.

continued on next page

IAB/IESG on Cryptographic Technology (*continued*)

Export controls and usage controls are slowing the deployment of security at the same time as the Internet is exponentially increasing in size and attackers are increasing in sophistication. This puts users in a dangerous position as they are forced to rely on insecure electronic communication.

Key size

It is not acceptable to restrict the use or export of cryptosystems based on their key size. Systems that are breakable by one country will be breakable by others, possibly unfriendly ones. Large corporations and even criminal enterprises have the resources to break many cryptosystems. Furthermore, conversations often need to be protected for years to come; as computers increase in speed, key sizes that were once out of reach of cryptanalysis will become insecure.

Public key infrastructure

Use of public key cryptography often requires the existence of a *Certification Authority* (CA). That is, some third party must sign a string containing the user's identity and public key. In turn, the third party's key is often signed by a higher-level certification authority.

Such a structure is legitimate and necessary. Indeed, many governments will and should run their own CAs, if only to protect citizens' transactions with their governments. But certification authorities should not be confused with escrow centers. Escrow centers are repositories for private keys, while certification authorities deal with public keys. Indeed, sound cryptographic practice dictates that users never reveal their private keys to anyone, even the certification authority.

Keys should not be revealable

The security of a modern cryptosystem rests entirely on the secrecy of the keys. Accordingly, it is a major principle of system design that to the extent possible, secret keys should never leave their user's secure environment. Key escrow implies that keys must be disclosed in some fashion, a flat-out contradiction of this principle. Any such disclosure weakens the total security of the system.

Data recovery

Sometimes escrow systems are touted as being good for the customer because they allow data recovery in the case of lost keys. However, it should be up to the customer to decide whether they would prefer the more secure system in which lost keys mean lost data, or one in which keys are escrowed to be recovered when necessary. Similarly, keys used only for conversations (as opposed to file storage) need never be escrowed. And a system in which the secret key is stored by a government and not by the data owner is certainly not practical for data recovery.

Signature keys

Keys used for signatures and authentication must never be escrowed. Any third party with access to such keys could impersonate the legitimate owner, creating new opportunities for fraud and deceit. Indeed, a user who wished to repudiate a transaction could claim that his or her escrowed key was used, putting the onus on that party. If a government escrowed the keys, a defendant could claim that the evidence had been forged by the government, thereby making prosecution much more difficult. For electronic commerce, non-repudiation is one of the most important uses for cryptography; and non-repudiation depends on the assumption that only the user has access to the private key.

Protection of the existing infrastructure

In some cases, it is technically feasible to use cryptographic operations that do not involve secrecy. While this may suffice in some cases, much of the existing technical and commercial infrastructure cannot be protected in this way.

**Conflicting
international policies**

For example, conventional passwords, credit card numbers, and the like must be protected by strong encryption, even though some day more sophisticated techniques may replace them. Encryption can be added on quite easily; wholesale changes to diverse systems cannot.

Conflicting restrictions on encryption often force an international company to use a weak encryption system, in order to satisfy legal requirements in two or more different countries. Ironically, in such cases either nation might consider the other an adversary against whom commercial enterprises should use strong cryptography. Clearly, key escrow is not a suitable compromise, since neither country would want to disclose keys to the other.

Multiple encryption

Even if escrowed encryption schemes are used, there is nothing to prevent someone from using another encryption scheme first. Certainly, any serious malefactors would do this; the outer encryption layer, which would use an escrowed scheme, would be used to divert suspicion.

**Escrow of private keys
won't necessarily allow
data decryption**

A major threat to users of cryptographic systems is the theft of long-term keys (perhaps by a hacker), either before or after a sensitive conversation. To counter this threat, schemes with *Perfect Forward Secrecy* (PFS) are often employed. If PFS is used, the attacker must be in control of the machine during the actual conversation. But PFS is generally incompatible with schemes involving escrow of private keys. (This is an oversimplification, but a full analysis would be too lengthy for this document.)

Conclusions

As more and more companies connect to the Internet, and as more and more commerce takes place there, security is becoming more and more critical. Cryptography is the most powerful single tool that users can use to secure the Internet. Knowingly making that tool weaker threatens their ability to do so, and has no proven benefit.

The Internet Architecture Board is described at:

<http://www.iab.org/iab>

The Internet Engineering Task Force and the Internet Engineering Steering Group are described at:

<http://www.ietf.org>

© Internet Society 1996. Reproduction or translation of the complete document, but not of extracts, including this notice, is freely permitted.

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our address is:

303 Vintage Park Drive

Foster City, California 94404-1138, USA

Phone: +1 415-578-6900 1-800-INTEROP

Internet: connexions@interop.com <http://www.interop.com>

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 610-892-1959 outside the USA.

This publication is distributed on an "as is" basis, without warranty. Neither the publisher nor any contributor shall have any liability to any person or entity with respect to any liability, loss, or damage caused or alleged to be caused, directly or indirectly, by the information contained in *ConneXions—The Interoperability Report*®

Call for Papers

NetWorld+Interop and the IEEE Communications Society will hold the *Engineer Conference on Broadband Access—Technologies, Systems and Services*, May 5–9, 1997 in the Las Vegas Convention Center. The goal of the conference is to provide a unique forum in the setting of NetWorld+Interop 97 for industries, universities, service providers, and end users to report on their latest advances in leading-edge broadband access technologies, and their impact on emerging enterprise and consumer mass markets. The broadband access technologies cover both wireline (e.g., HFC, FTTC/H, xDSL) and wireless (e.g., MMDS, LMDS, wireless ATM, direct broadcast and low earth orbit satellites) areas. Special emphasis will be placed on the integration of the two domains, and issues and solutions for large scale deployment and operations. This 2-day, 5-track conference intends to bring together state-of-the-art and original results in these exciting areas. The event is jointly sponsored by NetWorld+Interop and IEEE Communications Society Technical Committees on Cable-Based Delivery and Access Systems; Computer Communications; Enterprise Networking; Network Operations and Management; Personal Communications; and Transmission, Access and Optical Systems.

Topics

The Engineers Conference includes keynote speeches, panel discussions by leaders in the broadband access area, invited papers by recognized experts, and contributed papers by active researchers in the field. The program will cover the following topics:

- *Evolution of broadband access technologies and systems:* Invited papers or panel discussions on the trend of key access technologies by leaders in the field. Topics of discussion include recent advances in ADSL, HFC, FTTC(H), MMDS/LMDS, direct broadcast and low earth orbit satellites, as well as their technical and economic tradeoffs.
- *Advances in wireless access technologies:*
 - High-speed wireless modems and radio technologies
 - Media access/data link control and handoff protocols
 - Mobility and QoS management for broadband wireless access
 - Demand and mobility characterizations
 - Adaptive wireless access and multi-tier services
 - Security in wireless access services
- *System integration and operations:*
 - Wireless and wireline network integration
 - Wireless access and enterprise network integration
 - Signaling and control for mobile broadband systems
 - Management of multi-user, multi-service provider access
 - Network operations/management for multi-technology systems
 - Billing, regulatory, and economics
- *Congestion & Resource Management in Heterogeneous Networks:*
 - Bandwidth-on-demand solutions in heterogeneous environments
 - Routing and network resource management methods
 - Teletraffic engineering and performance modeling techniques
 - Access control, traffic shaping and enforcement
 - Traffic congestion and management
- *Video over Enterprise Networks:*
 - Schemes for picture encoding
 - Improvements in bandwidth utilization
 - Methods for accommodating variable latency
 - Multimedia integration
 - Standards progress

**Technical Program
Committee**

William Stephens
 Russell Hsing
 Curtis Siller
 Salah Aidarous
 Joseph Bannister
 Vijay Bhagavath
 Wai Chen
 Nim Cheung
 Bhumip Khasnabish
 David Kirsch
 Stan Moyer
 Gurudatta Parulkar
 Christian Rad
 Roberto Saracco
 Ya-Qin Zhang
 Doug Zuckerman

David Sarnoff Research Center (Chair)
 Bellcore (Vice Chair)
 Lucent Technologies (Vice Chair)
 Nortel Technology
 Aerospace Corporation
 Lucent Technologies
 Bellcore
 Bellcore
 GTE Labs
 NDC
 Bellcore
 Washington University
 AT&T
 CSELT
 David Sarnoff Research Center
 Paradyne

**NetWorld+Interop
representatives**

Dave Piscitello
 Liza Draper

Chair, N+I Program Committee
 Director of Content, Interop

Important dates

Deadline for contributed paper submission: October 1, 1996
 Notification of paper acceptance to authors: December 15, 1996
 Camera-ready papers due: February 15, 1997

Submission guidelines

Each paper must be in English and should not exceed 20 double-spaced, single-sided pages (12 point font, 26 lines per page), excluding figures. The title page of your submission must include: The name, affiliation, complete return address, e-mail, telephone and fax numbers of the author to whom all correspondence will be sent, as well as a 75–200 word abstract.

All other pages of the paper should contain the title of the paper, the name of the first author, and the page number. Five (5) copies of the paper should be forwarded to one of the submission addresses below. Electronic submission of papers (in *PostScript*) is encouraged.

Submission addresses

Dr. William E. Stephens
 David Sarnoff Research Center
 CN5300
 Princeton, NJ 08540-5300
 Tel: +1 609 734-3020
 Fax: +1 609 734-2049
 E-mail: wstephens@sarnoff.com

Dr. Russell T. Hsing
 Bellcore
 445 South Street, MCC-1A314R
 Morristown, NJ 07960
 Tel: +1 201 829-4950
 Fax: +1 201 829-5886
 E-mail: trh@bellcore.com

Dr. Curtis Siller
 Lucent Technologies
 1600 Osgood Street
 North Andover, MA 01845
 Tel: +1 508 960-1313
 Fax: +1 508 960-1477
 E-mail: csiller@bell-labs.com

CONNEXIONS

303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen

EDITORIAL ADVISORY BOARD Dr. Vinton G. Cerf
Senior Vice President, MCI Telecommunications
President, The Internet Society (1992 – 1995)

A. Lyman Chapin, Chief Network Architect,
BBN Communications

Dr. David D. Clark, Senior Research Scientist,
Massachusetts Institute of Technology

Dr. David L. Mills, Professor,
University of Delaware

Dr. Jonathan B. Postel, Communications Division Director,
University of Southern California, Information Sciences Institute



Printed on recycled paper

Subscribe to CONNEXIONS

U.S./Canada ☐ \$195. for 12 issues/year **All other countries** ☐ \$245. for 12 issues/year

Name _____ Title _____

Company _____ E-mail _____

Address _____

City _____ State _____ Zip _____

Country _____ Telephone () _____

Fax () _____

☐ Check enclosed (in U.S. dollars made payable to **CONNEXIONS**).

☐ Visa ☐ MasterCard ☐ American Express ☐ Diners Club Card# _____ Exp.Date _____

Signature _____

Please return this application with payment to:

Back issues available upon request \$15./each
Volume discounts available upon request

CONNEXIONS

303 Vintage Park Drive, Suite 201
Foster City, CA 94404-1138
415-578-6900 FAX: 415-525-0194
connexions@interop.com

CONNEXIONS